



Master Technologies de l'Hypermédia
Domaine : Sciences, Technologies, Santé
Mention : Informatique
Parcours : Technologies de l'Hypermédia (THYP)

Cahier des charges
Plateforme intelligente de gestion du processus de
Peer-Review académique

Réalisé par : BOUSSAID Amine
Encadré par : Professeur IMAD SALAH
Cours : Théories et conception des hypermédias

1. Présentation du Sujet du Projet

4 : 6

1. Introduction
2. Problématique
3. Solutions proposées
4. Intégration du Système de Recommandation
5. Conclusion

2. Analyse et Conception

7 : 17

1. Méthodologie adoptée
 - 2.1.1. Méthode Agile Scrum
2. Modélisation du projet
 - 2.2.1. Langage de modélisation
 - 2.2.2. Les acteurs du système
 - 2.2.3. Diagramme de cas d'utilisation
3. Description détaillée des cas d'utilisation
 - 2.3.1. Les cas d'utilisation communs entre les acteurs
 - 2.3.1.1. Inscription
 - 2.3.1.2. S'AUTHTIFIER
 - 2.3.2. Cas d'utilisation de l'Auteur :
 - 2.3.3. Cas d'utilisation de l'Expert
4. Modélisation des données
 - 2.4.1. Diagramme de classe
 - 2.4.2. Modèle conceptuel du système de recommandation
5. Fonctionnalités Détaillées

3. Mise en œuvre de l'application

18 : 23

1. Introduction à l'environnement de développement
2. Architecture MVC :
 - 3.2.1. Les différentes parties de l'architecture MVC
3. Environnement de travail en Modélisation
4. Environnement de travail
5. Architectures et langages de programmation utilisés
 - 3.5.1. Technologies utilisées
 - 3.5.1.1. Angular (Frontend)
 - 3.5.1.2. Node JS
 - 3.5.1.3. Express JS
 - 3.5.1.4. MySQL

CHAPITRE 1 :

Présentation du Sujet du Projet

1. Introduction
2. Problématique
3. Solutions proposées
4. Intégration du Système de Recommandation
5. Conclusion

1. Introduction

La validation de la recherche scientifique repose sur le processus rigoureux de la **revue par les pairs (peer-review)**. Cependant, ce processus est souvent entravé par des systèmes de gestion manuels, lents et peu transparents, ce qui impacte les délais de publication et la charge de travail des éditeurs.

Le projet **PeerConnect** vise à développer une plateforme web complète qui automatise et optimise le workflow de peer-review pour articles scientifiques et rapports. En s'appuyant sur les technologies de l'Hypermédia, PeerConnect cherche à apporter une solution professionnelle et moderne aux besoins des communautés académiques et des institutions de recherche.

2. Problématique

Comment concevoir et développer une plateforme hypermédia capable d'assurer une gestion fluide, rapide et de haute qualité de la revue par les pairs, en surmontant les défis suivants :

1. **Le délai** : Réduire le temps moyen nécessaire entre la soumission d'un article et la décision finale.
2. **L'efficacité du matching** : Assigner automatiquement et intelligemment les experts les plus compétents à chaque article soumis.
3. **La transparence et le suivi** : Offrir des outils de tableau de bord et de statistiques pour garantir la visibilité de la progression à tous les acteurs.

3. Solution

La plateforme sera une application web dynamique et responsive, construite autour des principes du développement moderne (architecture modulaire et orientée services).

1. **Workflow End-to-End Automatisé** : Gestion complète du cycle de vie de l'article (soumission, assignation, révision, décision) avec des rappels automatiques et une gestion de la cascade en cas de refus d'un reviewer.
2. **Interface de Review Enrichie** : Fourniture d'outils d'annotation PDF directement dans le navigateur pour un retour précis et contextuel des experts.
3. **Tableau de Bord et Statistiques** : Des vues personnalisées pour l'Administrateur (taux d'acceptation, délais moyens) et pour l'Expert (score de crédibilité, articles en cours).
4. **Sécurité et Confidentialité** : Implémentation des mécanismes de double-blind review pour assurer l'anonymat entre l'auteur et l'expert.

4. Intégration du Système de Recommandation

Le cœur de l'intelligence de PeerConnect réside dans son **Matching Algorithm**. Ce système d'aide à la décision sera intégré pour :

- **Analyser l'Article** : Extraire automatiquement les mots-clés et les domaines d'expertise à partir des métadonnées de l'article soumis.
- **Évaluer l'Expert** : Utiliser les données du profil de l'Expert (domaine d'expertise déclaré, historique de reviews, score de crédibilité) et sa disponibilité actuelle.
- **Proposer une Correspondance** : Calculer un score de pertinence pour recommander à l'Administrateur une liste classée des experts les plus qualifiés pour l'article en question. Ce système minimise l'effort manuel et accélère l'étape d'assignation.

Conclusion :

Le projet PeerConnect, en s'appuyant sur une architecture technique éprouvée (MERN Stack) et en intégrant un système de recommandation intelligent, vise à transformer un processus académique traditionnel en une expérience professionnelle, efficace et moderne. Ce cahier des charges servira de référence pour le développement des modules prioritaires de l'application.

CHAPITRE 2 :

ANALYSE ET CONCEPTION

1. Méthodologie adoptée

1.1 Méthode Agile Scrum

2. Modélisation du projet.

2.1 Langage de modélisation

2.2 Les acteurs du système

2.3 Diagramme de cas d'utilisation

3. Description détaillée des cas d'utilisation

3.1 Les cas d'utilisation communs entre les acteurs

3.1.1 Inscription

3.1.2 S'authentifier

3.2 Cas d'utilisation de l'Acteur :

3.3 Cas d'utilisation de l'Expert

4. Modélisation des données

4.1 Diagramme de classe

4.2 Modèle conceptuel du système de recommandation

5 Fonctionnalités Détaillées

1. Méthodologie adoptée

La méthodologie choisie est la **méthode Agile Scrum**, adaptée aux projets web dynamiques.

Elle repose sur des itérations courtes appelées *sprints*, permettant d'obtenir rapidement des versions fonctionnelles de l'application et d'intégrer facilement les retours.

1.1 Méthode Agile Scrum

Les méthodes agiles sont issues initialement du manifeste agile. Ce manifeste a été créé par des développeurs de systèmes informatiques qui, comme Jeff Sutherland, étaient fatigués de voir ses projets embourbés et sa créativité inhibée par des exigences bureaucratiques et des méthodes dépassées. En conséquence, ils ont revu complètement les méthodes de développement de logiciel. Cela implique entre autres :

- Des réponses rapides et flexibles face à des scénarios changeants.
- Des plans d'action flexibles.
- La priorité donnée au travail d'équipe collaboratif et rationnel.
- L'auto-organisation comme caractéristique essentielle de fonctionnement.
- Une méthode de développement incrémentale basée sur les retours utilisateurs.

Scrum est une méthode grâce à laquelle les gens peuvent aborder et résoudre de manière créative des problèmes complexes – et en constante transformation – afin de développer des produits ayant la plus grande valeur perçue pour le client final

2. Modélisation du projet

2.1 Langage de modélisation :

Le langage UML (Unified Modeling Language, ou langage de modélisation unifié) a été pensé pour être un langage de modélisation visuelle commun, et riche sémantiquement et syntaxiquement. Il est destiné à l'architecture, la conception et la mise en œuvre de systèmes logiciels complexes par leur structure aussi bien que leur comportement. Il peut également aider les développeurs à présenter leur description d'un système d'une manière compréhensible pour les spécialistes externes. UML est principalement utilisé dans le développement de logiciels orientés objet.

Figure 1 : Logo du Langage UML



2.2 Les acteurs du système :

Dans le contexte de PeerConnect, nous définissons trois acteurs principaux. Le système de votre ancien rapport, basé sur les "Clients" et les "Prestataires de Service", est remplacé par les rôles spécifiques au peer-review.

Acteur	Rôle dans PeerConnect	Responsabilités Principales
Auteur	L'utilisateur soumettant un travail pour évaluation.	Soumettre des articles (PDF + métadonnées), consulter le statut de l'évaluation, répondre aux révisions.
Expert / Reviewer	L'utilisateur chargé d'évaluer la qualité scientifique de l'article.	Accepter/Refuser les demandes de review, soumettre un rapport d'évaluation (notes, commentaires, annotations).
Administrateur / Éditeur	Le responsable du processus d'évaluation.	Gérer l'ensemble du workflow, assigner manuellement les experts (si nécessaire), consulter les statistiques globales, valider les profils des Experts.

2.3 Diagramme de cas d'utilisation :

Le diagramme de cas d'utilisation permet de recueillir, d'analyser, d'organiser les besoins et de capturer les fonctionnalités du système avec une vision orientée utilisateur. Son élaboration est une étape primordiale dans toute modélisation de système informatique. Pour chaque acteur, on précisera son diagramme de cas d'utilisation :

3. Description détaillée des cas d'utilisation :

Pour mieux comprendre le fonctionnement du système, nous allons détailler quelques cas importants :

3.1 Les cas d'utilisation communs entre les acteurs :

3.1.1 Inscription :

L'Inscription est une étape obligatoire pour tout utilisateur avant d'accéder à l'application.

Nom du cas	S'Inscrire
Acteur Principale	Tous les acteurs de l'application
Objectif	S'authentifier pour accéder aux services de l'application
Pré Condition	Aucune
Contraintes	Avoir une adresse email valide
Scénario normal	<ol style="list-style-type: none"> 1. L'utilisateur fournit les informations (nom, email, mot de passe). 2. Si l'utilisateur s'inscrit comme Expert, il fournit également ses domaines d'expertise et son CV. 3. Le système crée le compte. 4. Si l'Expert s'inscrit, le compte reste en statut 'En attente' jusqu'à validation par l'Administrateur.
Scénario D'échec	<ol style="list-style-type: none"> 1. Les champs sont vides ou incorrects 2. Le système affiche un message d'erreur
Post Condition	S'authentifier

Tableau 2 : Description détaillée du cas d'utilisation « S'inscrire »

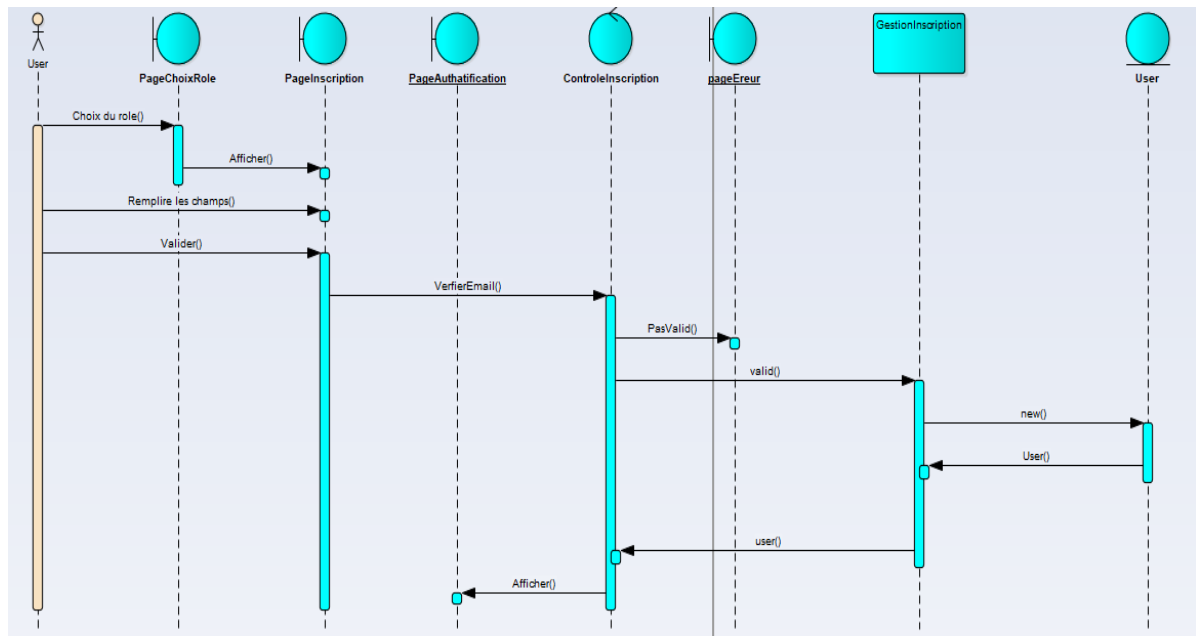


Figure 2:: Diagramme de séquence du cas d'utilisation « Inscription »

3.1.2 S'AUTHTENTIFIER :

Nom du cas	S'Authentifier
Acteur Principale	Tous les acteurs de l'application
Objectif	Accéder aux services de l'application
Pré Condition	Le compte existe. Le compte Expert est validé par l'Administrateur.
Contraintes	1.Respecter la casse du mot de passe 2.Avoir un profil actif
Scénario normal	1. L'utilisateur saisit son email et mot de passe. 2. Le système vérifie les informations d'identification. 3. L'utilisateur est redirigé vers son tableau de bord spécifique.
Scénario D'échec	1.Les champs sont vides ou incorrects 2. Le système affiche un message d'erreur
Post Condition	S'authentifier

Tableau 3 : Description détaillée du cas d'utilisation « S'authentifier »

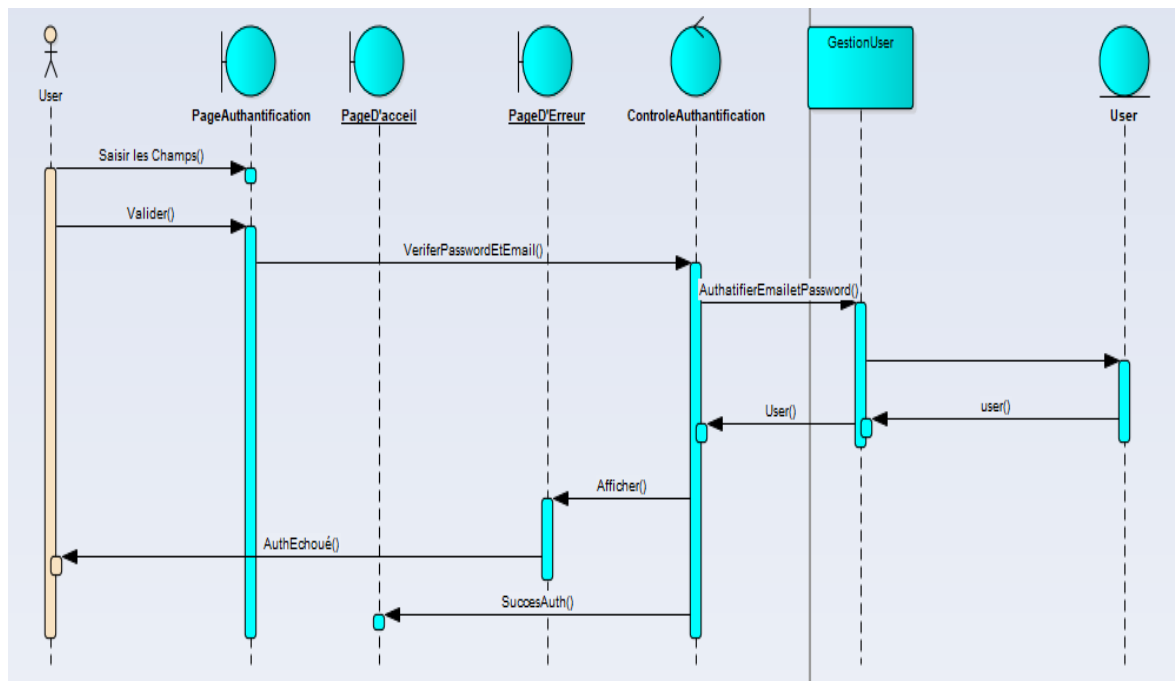


Figure 3:: Diagramme de séquence du cas d'utilisation « Authentification »

3.2 Cas d'utilisation du l'Auteur :

Le rôle de l'Auteur est centré sur la soumission de son travail et le suivi de son évaluation.

Nom	Soumettre un Article
Objectif	Déposer un nouveau document pour l'évaluation.
Description	L'Auteur télécharge le document (PDF) et renseigne les métadonnées obligatoires (titre, résumé, domaine, mots-clés).
Nom	Consulter le Statut de l'Article
Objectif	Suivre la progression de l'évaluation en temps réel.
Description	L'Auteur peut voir les étapes franchies (Assignment en cours, En review, Révisions demandées, Accepté/Rejeté).

Tableau 4 : Description détaillée du cas d'utilisation « l'Auteur »

3.3 Cas d'utilisation de l'Expert :

Le rôle de l'Expert est essentiel pour la validation scientifique.

NOM	GERER UNE DEMANDE DE REVIEW
OBJECTIF	Accepter ou décliner une nouvelle demande d'évaluation.
DESCRIPTION	L'Expert reçoit une notification et, après avoir consulté le résumé de l'article, décide d'accepter ou de refuser la tâche.
NOM	Réaliser la Review
OBJECTIF	Fournir une évaluation complète et structurée de l'article.
DESCRIPTION	L'Expert utilise l'interface enrichie (incluant l'outil d'annotation PDF) pour donner des notes sur des critères spécifiques (qualité, originalité, méthodologie) et fournir une recommandation finale.

Tableau 5 : Description détaillée du cas d'utilisation « l'Expert »

4. Modélisation des données :

4.1 Diagramme de classe :

Le diagramme de classes est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci. Ce diagramme fait partie de la partie statique d'UML car il fait abstraction des aspects temporels et dynamiques.

Une classe décrit les responsabilités, le comportement et le type d'un ensemble d'objets. Les éléments de cet ensemble sont les instances de la classe.

Une classe est un ensemble de fonctions et de données (attributs) qui sont liés ensemble par un champ sémantique. Les classes sont utilisées dans la programmation orientée objet. Elles permettent de modéliser un programme et ainsi de découper une tâche complexe en plusieurs petits travaux simples.

Les classes peuvent être liées entre elles grâce au mécanisme d'héritage qui permet de mettre en évidence des relations de parenté. D'autres relations

sont possibles entre des classes, chacune de ces relations est représentée par un arc spécifique dans le diagramme de classes.

Une classe est représentée par un rectangle séparé en trois parties :

- la première partie contient le nom de la classe.
- la seconde contient les attributs de la classe.
- la dernière contient les méthodes de la classe. La seconde et la dernière représentent le comportement de la classe.

DIAGRAMME DE CLASSE :

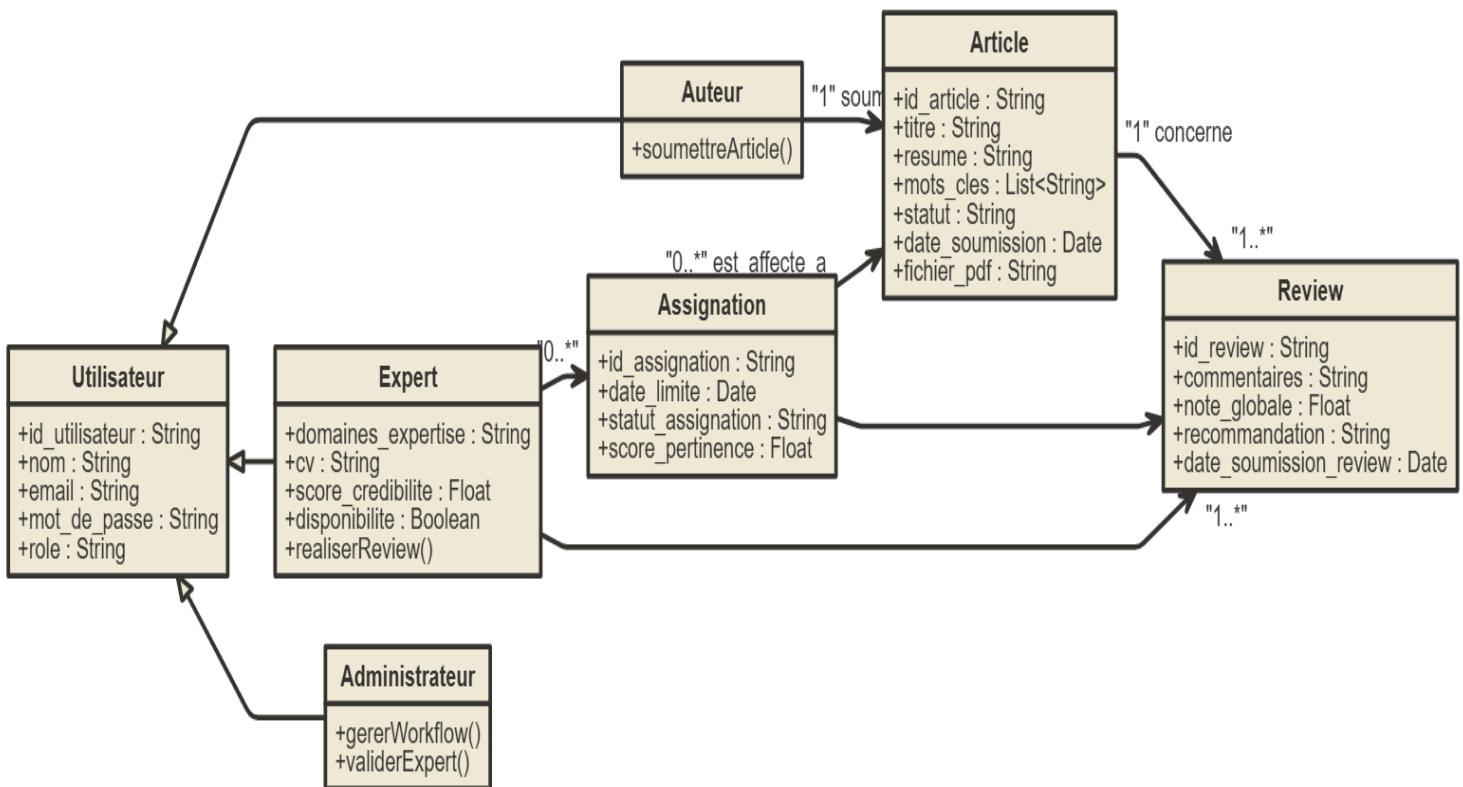


Figure 4: Diagramme de Classes

4.2 Modèle conceptuel du système de recommandation

Le Modèle Conceptuel du Matching Algorithme sera basé sur des relations pondérées entre les entités suivantes :

- **Article** : Caractérisé par un ensemble de **Mots-clés** et un **Domaine d'expertise**.
- **Expert** : Caractérisé par un ensemble de **Domaines d'expertise** (avec un niveau de maîtrise), un **Historique de Reviews**, et une **Disponibilité**.

Relation de Matching (Score) : Le système de recommandation calculera un **Score de Pertinence** pour chaque paire (Article, Expert) en utilisant une fonction qui combine :

Ce modèle permet d'avoir une structure de données optimisée pour l'exécution rapide de l'algorithme, ce qui est crucial pour l'efficacité de la plateforme.

5. Fonctionnalités Détaillées :

Pour valider le module académique, les fonctionnalités *Must-Have* (M) seront prioritaires. Les *Good to Have* (G) seront développées comme extensions professionnelles.

#	FONCTIONNALITE	ACTEUR(S)	PRIORITE	DESCRIPTION
F01	Soumission Article	Auteur	M	Permettre à un Auteur de télécharger un PDF et de renseigner les métadonnées de l'article.
F02	Sélection Expert	Administrateur	M	Permettre à l'Admin de choisir un ou plusieurs Experts pour un article.
F03	Notification Assignation	Expert	M	L'Expert est notifié et peut Accepter ou Refuser la demande de review.
F04	Workflow	Système	M	En cas de refus de l'Expert, le système assigne automatiquement l'article à

	Automatique			l'Expert suivant (ou notifie l'Admin).
F05	Saisie de Review	Expert	M	Permettre à l'Expert d'ajouter un avis textuel, une note globale et une recommandation finale (Accept/Reject...).
F06	Rapport Compilé	Auteur	M	L'Auteur reçoit un rapport final regroupant les avis et la décision de l'éditeur.
F07	Dashboard Utilisateur	Tous	G	Affichage de l'état des articles (Auteur) ou des articles à revoir (Expert).
F08	Matching Algorithme	Administrateur	G	Le système propose les meilleurs Experts pour un article donné sur base d'un algorithme de similarité.
F09	Annotation PDF	Expert	G	Fournir un outil dans le navigateur pour surligner et ajouter des commentaires contextuels dans le PDF.
F10	Gestion Confid.	Administrateur	G	Définir le mode de review (simple-blind ou double-blind).

Conclusion :

Ce chapitre a été consacré à différentes étapes de la conception détaillée. Du raffinement du diagramme de cas d'utilisation, vers du diagramme de séquence de plusieurs cas d'utilisations pour montrer les interactions entre les acteurs et le système. Finalement on a terminé avec le diagramme de classe avec lequel on a présenté la structure de notre application.

Chapitre 3 :

Mise en œuvre de l'application

(Architecture et Technique utilisées)

1. Architecture MVC

1.1 Les différentes parties de l'architecture MVC

2. Environnement de travail en Modélisation :

3. Environnement de travail

4. Architectures et langages de programmation utilisés

4.1 Technologies utilisées

- a) Angular
- b) Node JS
- c) Express JS
- d) MySQL

Introduction

En programmation informatique, un environnement de développement est un ensemble d'outils qui permet d'augmenter la productivité des programmeurs qui développent des logiciels. Son objectif est d'augmenter la productivité des programmeurs en automatisant une partie des activités et en simplifiant les opérations. Ils visent également à améliorer la qualité de la documentation en rapport avec le logiciel en construction. Certains environnements de développement offrent également la possibilité de créer des prototypes, de planifier les travaux et de gérer des projets. Du coup, après avoir achevé l'étape de conception de l'application, on va se concentrer dans ce chapitre sur l'environnement de développement, en présentant les outils de développement et les différentes technologies utilisées.

1. Architecture MVC :

Modèle-vue-contrôleur ou MVC est un motif d'architecture logicielle destiné aux interfaces graphiques lancé en 1978 et très populaire pour les applications web. Cette architecture permet de bien organiser son code source. Elle aide à savoir quels fichiers créer, mais surtout à définir leur rôle. Le but de MVC est justement de séparer la logique du code en trois parties que l'on retrouve dans des fichiers distincts.

1.1. Les différentes parties de l'architecture MVC :

Modèle-vue-contrôleur est composé de trois types de modules ayant trois responsabilités différentes, comme le nom l'indique : les modèles, les vues et les contrôleurs.

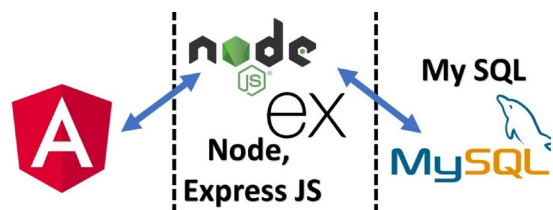


Figure 5 :
Architecture MVC

Modèle : Gère la logique métier et l'accès aux données. Il représente les entités (comme Article, Utilisateur, Review) et les règles qui les régissent. C'est l'interface avec la base de données MySQL.

Vue : Responsable de l'affichage des données à l'utilisateur (interface graphique). Dans notre cas, il s'agira de l'interface React.js.

Contrôleur : Reçoit les requêtes de l'utilisateur (via la Vue), interagit avec le Modèle pour mettre à jour l'état du système, puis sélectionne la Vue appropriée pour la réponse. Dans notre cas, le Contrôleur sera géré par **Express.js** et **Node.js**.

2. Environnement de travail en Modélisation :

Enterprise Architect est un logiciel de modélisation et de conception UML, édité par la société australienne 'SparxSystems'. Couvrant, par ses fonctionnalités, l'ensemble des étapes du cycle de conception d'application, il est l'un des logiciels de conception et de modélisation les plus reconnus

*Figure 6:
Logo de Enterprise Architect*



3. Environnement de travail :



Visual Studio Code : est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS3.

Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code (IntelliSense4.), les snippets, la refactorisation du code et Git intégré. Les utilisateurs peuvent modifier le thème, les raccourcis clavier, les préférences et installer des extensions qui ajoutent des fonctionnalités supplémentaires

4. Architectures et langages de programmation utilisés :

4.1. Technologie utilisés :

a) Angular:



Angular est le framework de choix pour le front-end. Il permet de construire des applications web monopages (Single Page Applications - SPA) hautement structurées et maintenables, idéales pour les tableaux de bord dynamiques et les outils d'interface utilisateur complexes requis par PeerConnect.

***Figure 7:
Logo de React***

b) Node.js :



***Figure 8:
Logo de Node.js***

Node.js est une plateforme logicielle libre en JavaScript, orientée vers les applications réseau évènementielles hautement concurrentes qui doivent pouvoir monter en charge. Elle utilise la machine virtuelle V8, la bibliothèque 'libuv' pour sa boucle d'évènements, et implémente sous licence MIT les spécifications CommonJS. Parmi les modules natifs de Node.js, on retrouve http qui permet le développement de serveur HTTP. Ce qui autorise, lors du déploiement de sites internet et d'applications web développés avec Node.js, de ne pas installer et utiliser des serveurs 'webs' tels que Nginx ou Apache. Concrètement, Node.js est un environnement bas niveau permettant l'exécution de JavaScript côté serveur.

c) Express.js :

Express.js est un framework pour construire des applications web basées sur Node.js². C'est de fait le framework standard pour le développement de serveur en Node.js³. L'auteur original, TJ Holowaychuk, le décrit comme un serveur inspiré de Sinatra⁴ dans le sens qu'il est relativement minimaliste tout en permettant d'étendre ses fonctionnalités via des plugins.



Figure 9:
Logo de Express.js

d) MySQL :

MySQL est sélectionné comme système de gestion de base de données relationnelle (SGBDR).

Il est robuste, mature et particulièrement adapté pour :

- Assurer l'intégrité des données grâce aux transactions SQL.
- Gérer les relations complexes et structurées (Article, Expert, Review, Assignment) via des schémas stricts, essentiels pour un workflow académique critique.



Figure 10:
Logo de MySQL

Conclusion :

Après avoir terminé l'étude technique, ce chapitre a mis l'accent sur la description des caractéristiques de l'environnement du travail et décrit les plateformes matérielles et logiciel sur lesquelles on a construit notre application.

Conclusion Générale

Le projet "PeerConnect" représente une application concrète des Technologies de l'Hypermédia, en utilisant une architecture robuste et en intégrant une dimension intelligente (système de recommandation) pour résoudre une problématique réelle et complexe du milieu académique. Ce CdC sert de feuille de route pour le développement et la validation du module.